U.S. NAVAL ACADEMY
COMPUTER SCIENCE DEPARTMENT
TECHNICAL REPORT

Mobile Networks in IPV6

Noronha, Sean

USNA-CS-TR-2009-01

January 3, 2009

USNA Computer Science Dept. ○ 572M Holloway Rd Stop 9F ○ Annapolis, MD 21403

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **03 JAN 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Mobile Networks in IPV6**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Naval Academy,Computer Science Department,572M Holloway Rd Stop 9F,Annapolis,MD,21403**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**21**	

U.S. NAVAL ACADEMY
COMPUTER SCIENCE DEPARTMENT
TECHNICAL REPORT

Mobile Networks in IPv6

Noronha, Sean J.

USNA-CS/IT

Computer Science Department
SI495A: Research Project Report
Fall AY09

Mobile Networks in IPv6

by

Midshipman Sean Noronha, 094788

United States Naval Academy
Annapolis, MD

Date

Asst Prof. Thomas Augustine, LTCOL, USAF
Department of Computer Science

Date

Department Chair Endorsement

Thomas Logue, CAPT, USN
Chair, Department of Computer Science

Date

Table of Contents

Executive Summary 4
Objectives and Expectations 5
Methodology 5
Study 1 5
Initial Setup 6
Setting the Baseline 7
Conducting the Tests 10
Study 1 - DataTransfer 10
Results 11
Study 2 – IPv4 to IPv6 Tunnel 11
Testing 15
Conclusions 16
Lessons Learned 18
Future Research Opportunities 18
Personal Experiences 19
Works Cited 20

Executive Summary

This research study, *Mobile Networks in IPv6*, was conducted at the United States Naval Academy (USNA) with the end goal of testing the capabilities of IPv6 in low bandwidth networks. Specifically, this study focused on the effects of using IPv6 on networks whose speeds are comparable to 56K Dial Up access, ISDN, and various other low bandwidth wireless networks. This research also ventured into the world of Voice over IP (VoIP) and analysis of the network traffic that it creates, but unfortunately time and complexity did not enable the study to start.

While the Office of Management and Budget (OMB) directed the Department of Defense (DoD) to complete the transition to IPv6 by 2008 (Domagalski 2007), there is still a great deal of testing to be accomplished before that goal is met. While most shore based DoD sites have excellent high bandwidth connections, there are still quite a number of units which do not have that luxury. The units in question, deployed ships, Marine ground units, and various others in a war zone, rely on low speed, high latency satellite or wireless connections to maintain connectivity to their commanders.

The first segment of this research project aimed to examine the feasibility of sending large amounts of data from high to small bandwidth networks. A significant portion of this project aimed to correlate MTU settings and transmission speeds. A hypothesis to be tested aimed to see if increasing or decreasing the MTU size would affect those speeds, but enough data was collected which showed that changing the MTU size only affected error rates, not transmission speeds. Tests were conducted to determine how line conditions would affect the way IPv6 would be transmitted across networks of varying bandwidth but extensive tests were unable to be conducted since legacy hardware was not able to be acquired. To help mitigate the hardware gap, a three-pronged approach of theory, simulation, and experiment setup was used to validate results.

The second segment of this research project focused on the viability of incorporating separate IPv6 networks tunneled through existing IPv4 connections. Our recommendation to have a tunneled network is derived from the impracticality of completely transitioning over to IPv6 without having a “stepping stone” in case there are design flaws or the lack of personnel to support such a colossal move. The research shows that this is a more palatable concept for future development as only simple knowledge of implementing a tunnel and appropriate hardware compatibility are required.

While it is very possible that IPv6 will eventually replace IPv4 as the protocol of choice, it is readily apparent that the migration process will be slow and somewhat painful. In order to ease the transfer process, networks should be setup using the tunneling method in order to fully train all personnel involved. This will drastically help mitigate the large scale chaos which would have ensued if IPv6 came online with no support for the managers.

Objectives and Expectations

1. Research the effects of MTU fluctuations in “low bandwidth” networks
2. Research QoS in order to gain an understanding of proposed implementation in IPv6
3. Research IPv6 address auto configuration to enable “Plug and Play” networks
4. Research the feasibility of using IPv6 to IPv4 tunneling to lessen the burden of migration from IPv4

Methodology

The table below shows the different tests that were slated to be conducted throughout this research study.

Test	**Description**
Networking Speed	Test varying network conditions using an IPv6 network. Modify variables such as MTU, full/half duplex, and different mediums (eg, Wireless, 10Mb, 4Mb, 2Mb links).
Tunneling	Test to see how tunneling works with different hardware and feasibility of using this capability for existing network infrastructure.

Study 1

The first segment of this study was to research the effect of IPv6 and its ability to handle finite bandwidth environments such as those found in dialup, early wireless, and ISDN type networks. Small bandwidth networks are the types of networks one would expect to find in thousands of small businesses across the globe. The military, while employing a wide variety of links, incorporated numerous finite capacity networks in its vast network. These connections offer connectivity to deployed forces in the far regions of the world. Ships often use those connections at sea as their only data link to the defense network.

Of the many interesting specifications of IPv6, the automatic MTU discovery (Cisco Systems 2003) is one of the more interesting concepts. While simple sounding in nature, this small design change many implications for networking. The result of this is the ability to send information at faster speeds, reducing latency, and in turn reducing operating costs. Given the level of difficulty in acquiring some of the required hardware, another avenue will be pursued.

The network consisted of two segments thus far: Client machines connected to a hub and a server with the ability to file share. The first segment ran on a 10Mb network connection while the second ran on a 100Mb segment. While this alone will not help thoroughly simulate the conditions of a “mobile” environment, it is part of the larger picture. After the initial setup is complete there will be a number of changes instituted to imitate low bandwidth conditions.

These include limiting the MTU on the client machines at varying levels, changing the specific channel configuration through half or full duplex, and using an early iteration of wireless technology.

Initial Setup

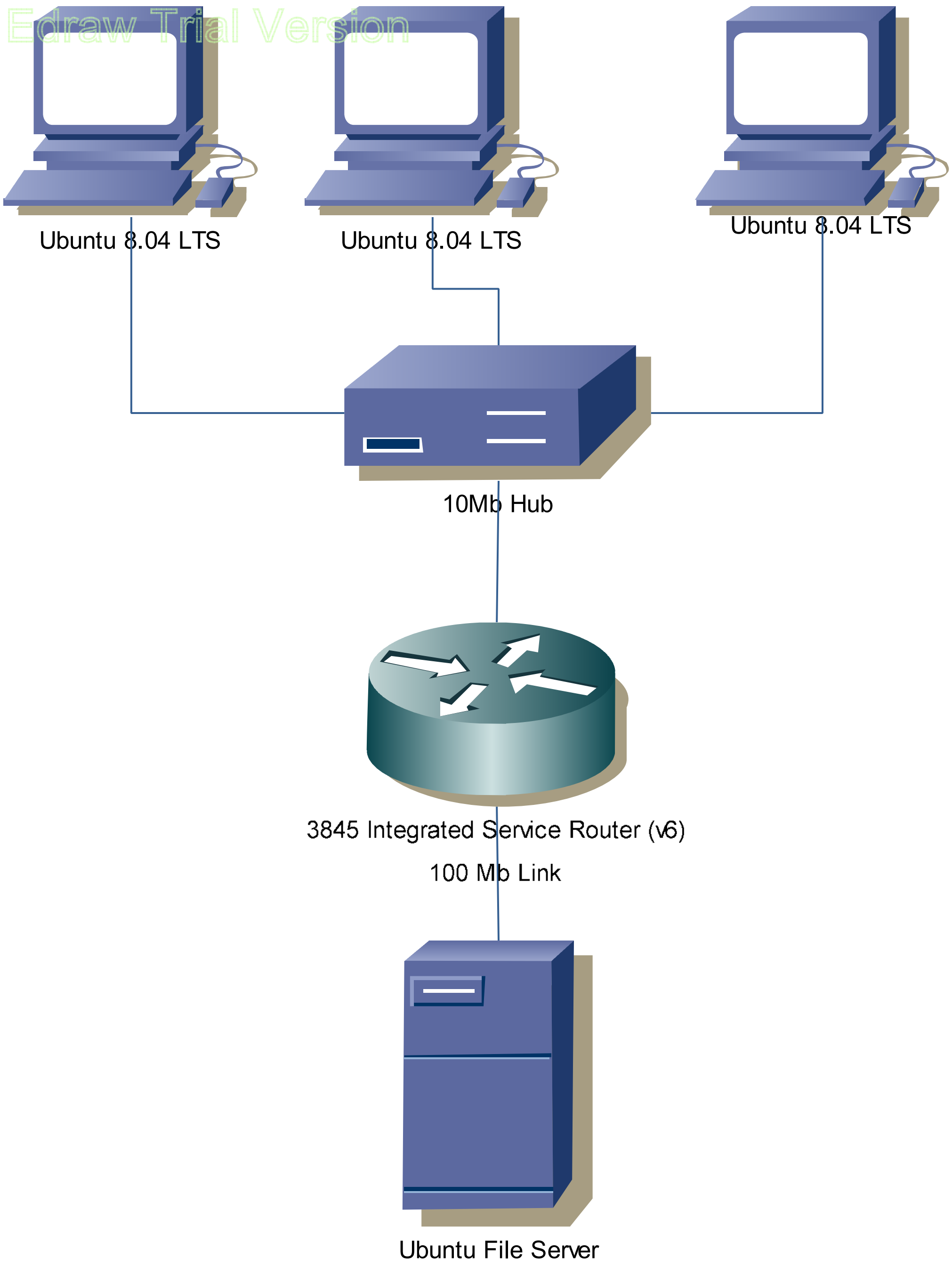


Figure 1

In this experiment, there are three computers running Ubuntu 8.04 LTS, dual-booted with Windows XP Pro SP2. The computers were all updated with the latest updates from the repositories.

As far as using Linux, specifically Ubuntu, the last study conducted by now ENS Josh Domagalski proved the versatility and dependability of running IPv6 on Linux boxes versus their Windows counter parts (Domagalski 2007). A later experiment will incorporate a single machine running Windows Vista SP1 and explore the claim that Vista is "IPv6 capable". Ubuntu was chosen as the distribution of choice due to its level of support, both from the community and the development team. It has quickly become a relatively easy OS to transition from Windows to a Linux environment. Since this project will effectively study the ease of setup and use for fleet personnel, using Ubuntu in the GNOME desktop environment helped make setup that much simpler.

The last piece of this puzzle, as far as computers go, was the addition of an Ubuntu server. This will function to serve large files and other resources which will be used in the testing environment. The server was supported by a 100Mb connection leading directly to the Cisco 3845, simulating a high speed connection. The second segment, with the other machines connected, was connected through a 10Mb connection to a centralized hub. The hub affords an opportunity to inspect all packets going in and coming out of that segment. This enables programs like Wireshark a much easier time to sniff packets. Switches were ruled out as a possibility since it would have taken more configuration setup to enabling the broadcast of traffic on a specific link in order to "see" all packet traffic.

Setting the Baseline

The centerpiece of the entire first group of experiments revolves around having a baseline for all testing. This baseline will provide us an idea of where we stand within the experiments. We are going to establish three types of baselines: Theoretical, Simulated, and Experimental.

The theoretical baseline formed by the math for a given scenario. Computing the maximum bandwidth, throughput, and various other variables are all included in this step. Simulated baselines are much more difficult to create, but thankfully we have a tool which is perfect for these environments. IT Guru Academic Edition from OPNET enables you to create almost any type of environment in the networking fields. You can simulate conditions found in an office, building, country, or even world scale environment. Drag and drop devices into the palette, make some connections, define variables and the program does the rest. Simulation time can be set to any variable, ranging from a quick burst of network to hours of testing. This software can replicate the scenarios extremely quickly.

For example, a simulation time of three hours means approximately five seconds of run time on the CPU. Then, you can view individual statistics from the session that range from bits received, latency, even network load. Of course these values are simulated, we were under the assumption that these simulations are as close to real life situations as possible with a computer.

Figure 2 shows the layout of our lab with all the equipment we are using.

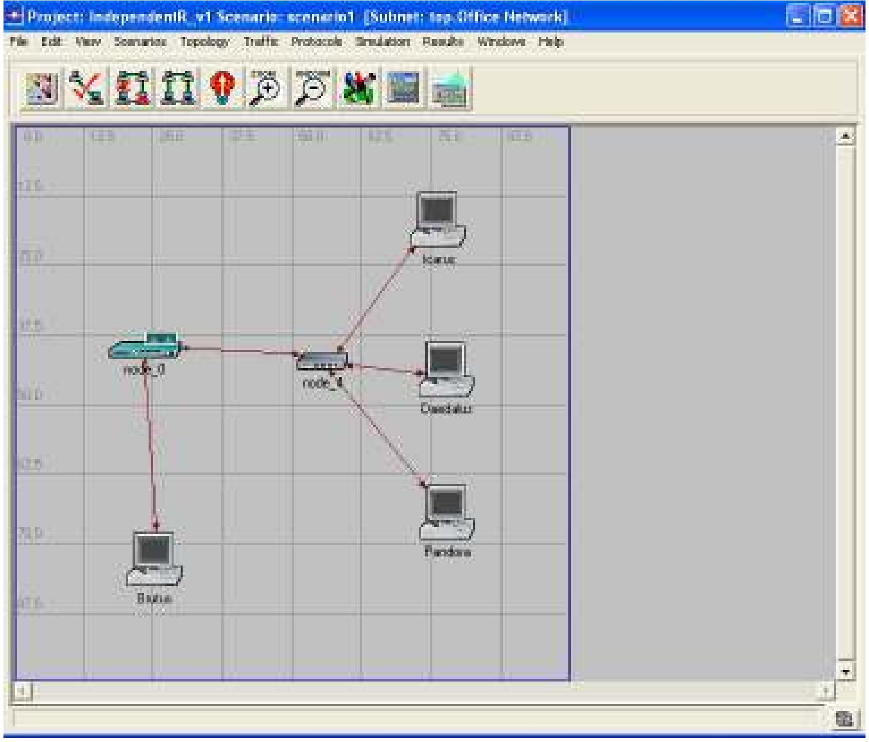

Project: IndependentR_v1 Scenario: scenario1 [Subnet: top.Office Network]
File Edit View Scenarios Topology Traffic Protocols Simulation Results Windows Help
Icarus
node_0
node_1
Daedalus
Pandora
Brutus

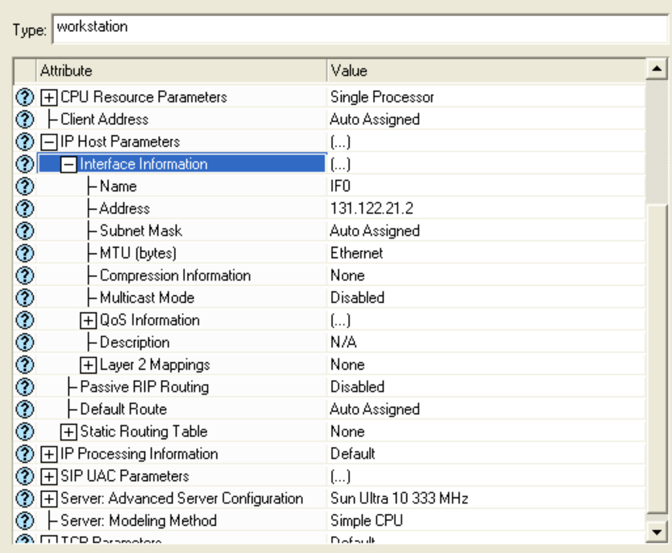
Figure 2

As evident from the image, we have our Cisco 3845 router connected to our main server as well as the "remote" machines. The server connection is specified as 100Mb, while the other connections are all 4Mb and connected to a hub. For our baseline simulation, each machine has been assigned an IPv6 address. The workstations are all running Sun Solaris, which is the closest OS substitution that was available. Since the kernels are alike in many ways, they should perform with similar results.

For this experiment, we are assuming that there are no other background processes running which would eat up CPU time. The image in Figure 3 shows the current configuration of each machine. As you can tell, the "IP Host Parameters" have the basic standard configuration that a normal machine would have. We will modify these settings for future baselines.

In the first baseline test, we will be looking at measuring throughput (bits/sec) in both the up and down stream directions. Unfortunately, this version of OPNET does not have IPv6 support. For now, we used the IPv4 addressing scheme to get an idea of where we stood. This enabled us to get a baseline for version 4. The baseline would be compared to the same simulation run under version 6 conditions. Data recovered from these two tests would allow us to see if any improvements made to version 6 had any effect on data transfer.

Figure 3 displays the configuration of our test network while Figure 4 shows the checked options of the many available for analysis.



Type: workstation

Attribute	Value
⊞ CPU Resource Parameters	Single Processor
├ Client Address	Auto Assigned
⊟ IP Host Parameters	(...)
⊟ Interface Information	(...)
├ Name	IF0
├ Address	131.122.21.2
├ Subnet Mask	Auto Assigned
├ MTU (bytes)	Ethernet
├ Compression Information	None
├ Multicast Mode	Disabled
⊞ QoS Information	(...)
├ Description	N/A
⊞ Layer 2 Mappings	None
├ Passive RIP Routing	Disabled
├ Default Route	Auto Assigned
⊞ Static Routing Table	None
⊞ IP Processing Information	Default
⊞ SIP UAC Parameters	(...)
⊞ Server: Advanced Server Configuration	Sun Ultra 10 333 MHz
├ Server: Modeling Method	Simple CPU
⊞ TCP Parameters	Default

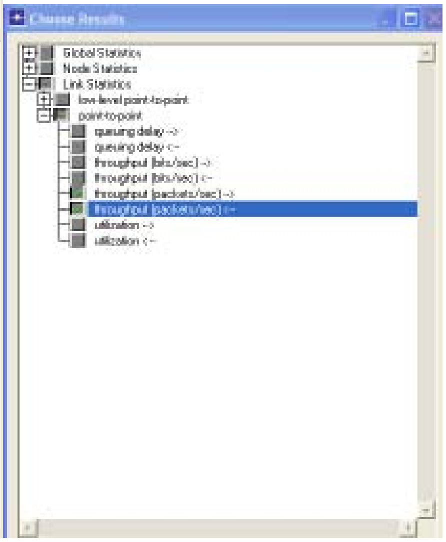


Figure 3 **Figure 4**

The established theoretical baseline for throughput on the various connections is as follows:

1. 10Mbps = 1.19Mbytes/sec
2. 100Mbps = 11.9 Mbytes/sec
3. 1000Mbps = 119.2 Mbytes/sec

Although the 1GB connection will not be tested, you can see from the numbers that there is obviously a pattern which shows the increase from each level on an order of magnitude per step. These numbers do not include the header overhead and simply assume that each link uses 100% of the bandwidth available.

We are also going to establish the baselines for the file sizes that will be tested for throughput testing:

1. Small File: 1MB
2. Medium File: 100MB
3. Large File: 4GB

The small file is to simulate a JPEG picture of medium to medium-high resolution while the medium file can be seen as either a small video clip or something else that is comparable. The large file is completely different. This is to meant to simulate high quality video content similar to the size you would find on a DVD which has been converted to MPEG or AVI.

As mentioned before, we also have to set the simulation baselines for the network. What we are looking for are “Global Statistics” and “Link Statistics” for

1. Ethernet Delay (entire network)
2. Throughput (bits/sec) (on links)
3. Traffic Received (bits/sec) (on Nodes)

In the first test, we will use an average traffic speed of 200Kbps. This is from an average packet size of 100 bytes/packet * 8 bits/byte * 1 packet/0.004 sec[1]. This is considered to be a “Low

[1] 0.004 refers to the propagation rate

Load" scenario. We will test a "High Load" scenario in the upcoming tests. The "High Load" scenario has an average traffic speed of 1 Mbps. This is computed from an average packet size of 500 bytes/packet * 8 bits/byte * 1 packet/0.004 sec.

Conducting the Tests

Study 1 - DataTransfer

Since we need to figure out a reference baseline for actual testing, creating files with sizes which adhere to the constraints listed above will help greatly in this process. A handy utility, *fsutil*, helped create files of sizes which you set. The utility was very easy to use. For example, creating a 4GB file was accomplished by using this command:

```
Fsutil file createnew <filename> size
```

The size was computed by using the following for the example:

```
Fsutil file createnew fourGB 4294967296
```

The file contained random data which was used to mimic data which could be encountered in an actual video file. For the tests, the timer and throughput values that were used came from the OS. The "System Monitor" tool available in Ubuntu determines, in real time, the throughput for the entire duration of the transfer. A simple timer was used with the experiment to start and stop when the transfer was complete. The test results are shown below:

1. Small File: .567 seconds, 1.76 Mbytes Average Throughput
2. Medium file: 17.78 seconds, 5.62 Mbytes Average Throughput
3. Large File: 12 minutes, 8.34 seconds, 5.23 Mbytes Average Throughput

While the numbers are very helpful in setting up a baseline, they did not prepare me for the problem which would ultimately stop the tests from happening.

The dilemma in question arose from the lack of hardware support for this research project. Today, it is very difficult to gain access to components which are legacy systems. This is true of networking components where most, if not all, of the older equipment is discarded upon the acquisition of upgraded systems. Various searches throughout the online forums revealed that the lowest possible link speed was 10Mb. Since this was already part of the lab setup, it quickly became evident that looking for changing MTU packets would not be possible given the link speeds.

Even when we change the values of half-duplex or full-duplex, propagation in circuitry happens so fast given the level of computation possible today, you will not notice any differences in your throughput. Note that it is not possible to combine a network link which is full-duplex with one that is half-duplex. Both links have to "talk" using the same duplex settings, which is why most computers are set to "Auto".

Changing the MTU settings on the machines would not help simulate smaller links either. While this would limit the number of packets sent and received, this is only a matter of software control and does not affect the physical properties of the line. Theoretically, while this would limit the packets, and subsequently decrease throughput, you would end up dropping the packets. Those same packets would be sent again, if reliable, and the throughput would be the same.

In essence, it was not physically possible to duplicate a small network environment given the level of constraints presented. Quite possibly, given a larger network over a wide distance, it would be possible to see dynamic changes in MTU in individual packets. Again, this is only achievable when you have multiple routers over varying connections between two end points. Measuring performance metrics, even Quality of Service ("QoS"), would be only possible given the right conditions.

One of the mentioned features of IPv6 was QoS. In fact, this has not and most probably will not be newly implemented over the QoS feature set you find in IPv4 (Cisco Systems 2003). According to an internal Cisco document, there is "No difference on protocols and methods to do QoS in IPv6 and IPv4" (Systems 2003), so one could expect the same levels of QoS in IPv6 that are in IPv4, at least theoretically.

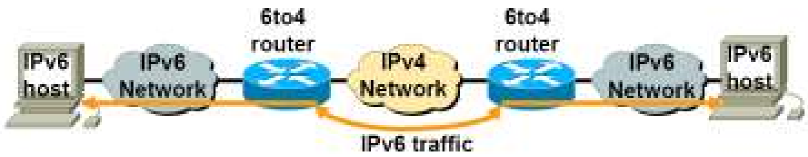
Results

Below are the results of the tests. All file transfer times are shown in seconds. Total time was monitored using a stopwatch and backed up by the transfer time shown at the completion of the file transfer. Any discrepancies between the computer and stopwatch were averaged[2].

File Size	Full-Duplex	Half-Duplex	Default MTU	Half MTU
Small File	.5 seconds	.5 seconds	.5 seconds	.5 seconds
Medium File	17.7 seconds	17.9 seconds	17.7 seconds	17.9 seconds
Large File	727.5 seconds	733.1 seconds	727.6 seconds	736.2 seconds

Study 2 – IPv4 to IPv6 Tunnel

Since it became obvious that there are still numerous issues in the integration of IPv6 we chose to test IPv6 over IPv4 as a viable alternative. The use of tunneling information through IPv4 networks makes having an IPv6 network possible, given that the internet has not switched over to a purely v6 environment. Tunneling with IPv6 is, quite simply, encapsulating information in the payload field of an IPv4 packet and sending that across the network with the intent of reversing the process if the destination is an IPv6 network.



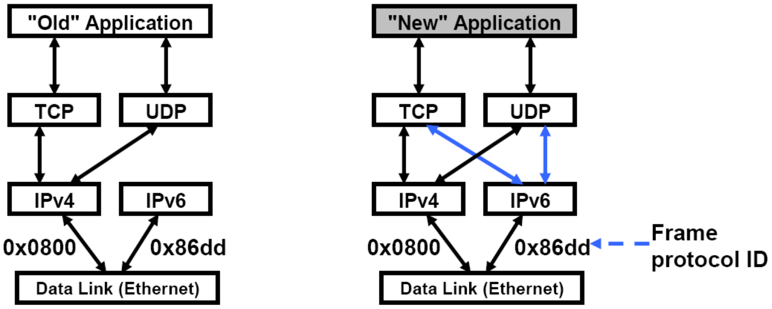
(Cisco Systems 2003)

[2] These tests were run multiple times to ensure accuracy

Figure 5

As you can see in Figure 5, the hardware requirements are sparse for tunnel establishment. There are some limitations to this process, most of which is the router of choice. Not every router supports this feature, either due to the IOS version being dated or the lack of onboard memory. While it is possible to run a router translation with relatively small amounts of memory, it is noted that this will drastically reduce the speed of the link, thus affecting performance.

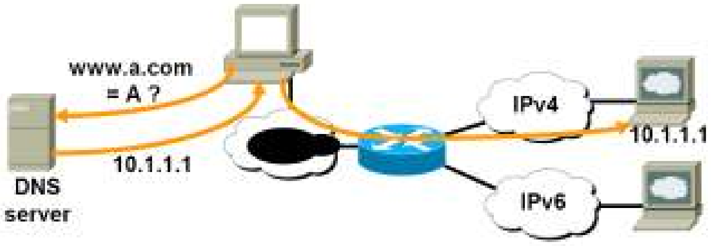
There are several methods to transition over to IPv6 seamlessly. One of the easier methods is a "Dual Stack" case where the IPv6 protocol is implemented on the host machines vice on the routers. In this case, both IPv6 and IPv4 stacks are enabled. Host applications can "talk" to both stacks without the user realizing that this is going on in the background. The only determining factor of which stack to use is based on the name lookup and application preference. (Figure 6)



(Cisco Systems 2003)

Figure 6

In a scenario where there is not an IPv6 applications (Figure 7)[3], the network works through the IPv4 as we are currently used to and connects to the IPv4 address.

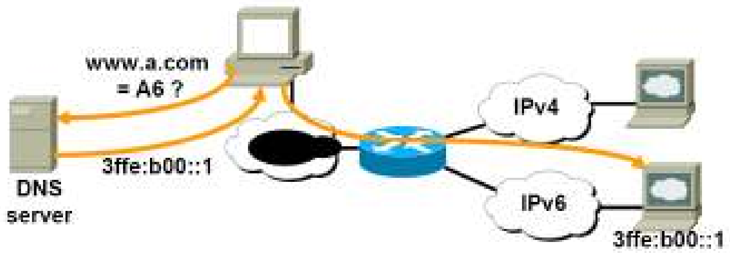


(Cisco Systems 2003)

Figure 7

In a scenario where there IS a request for an IPv6 enabled application or stack, the request goes through the same DNS server and requests and connects the host to the IPv6 address. (Figure 8)

[3] The "blob" in both Figure 7 and Figure 8 represent the "Internet Cloud"



(Cisco Systems 2003)

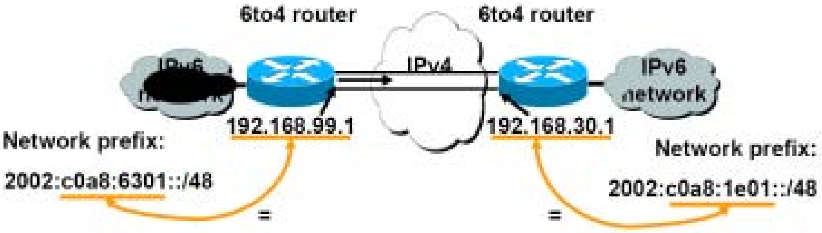
Figure 8

The only “catch” to the entire process is the router that is used on a “Dual Stack” network MUST be capable of implementing the encapsulation through the feature set of the IOS. There also has to be a single interface which has both an IPv6 and an IPv4 address, otherwise, this would not work.

As mentioned earlier, another method to use would be an “Overlay Tunnel”, which is in fact a fancy word for encapsulation. There are numerous ways of implementing this feature, ranging from manual configuration, semi-automated, automatic, 6to4, and 6over4.

Manual configuration relies on the user to configure the routers interface with an IPv4 and IPv6 address. The user also has to configure the routing protocol, link metrics, and various other preferences, whereas the automatic takes care of all of the above. Automatic relies on using a router generated IPv6 address which takes into account the MAC address to make up a portion of the hosts address. This is obviously not a recommended approach, but can be used to establish connectivity, possibly in small home user networks. Think of this as a “Plug and Play” approach which is much touted in various forms of the “UPNP” standard.

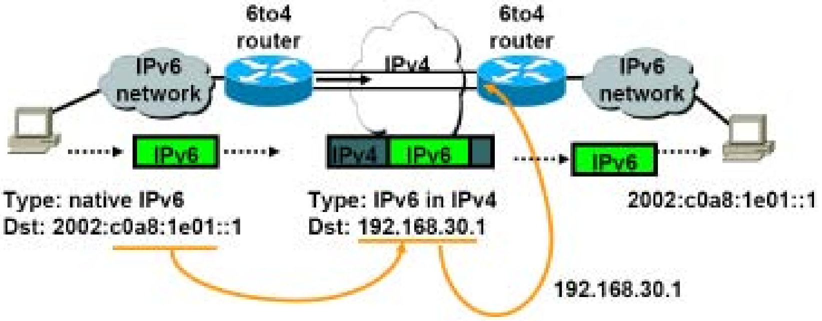
In the “6to4” method, we rely on the IPv4 protocol to help us reference addressing for our IPv6 network (Figure 9). In this scenario, the IPv4 address is translated and integrated into an IPv6 address. The tunneling accomplished in this case is automatic and the router IOS takes care of the translation.



(Cisco Systems 2003)

Figure 9

There is almost no difference in this method than what is accomplished with an IPv6 to IPv4 tunnel with the exception of incorporating the IPv4 address with a global IPv6 address. The same encapsulation is created and the exact same translation occurs. (Figure 10)[4]



(Cisco Systems 2003)

Figure 10

[4] Figure 10 shows the encapsulating process which happens in a typically IPv6 to IPv4 tunneled network. The encapsulation takes place on the "dual stack" router which has interfaces with both an IPv6 and an IPv4 address.

Testing

The figure below details the layout of the tunneled network[5]. The network starts with the 3845 Series router and continues through the 2600 Series routers, finally looping back to a separate interface on the 3845. Also attached to the 3845 is a computer to test connectivity to the interfaces via ICMP ping6 commands. The Cisco 3845 is running a pure IPv6 network with the latest Cisco IOS. This is connected to an interface on the 2600 which is running a split Ipv6/IPv4 network. All translation between V6 and V4 is taken care of by the IOS. The only catch is the requirement of having a Cisco IOS of 12.3 or greater. With this requirement an added memory requirement is also needed as running a split protocol on a 2600's meager 32MB of memory is rather slow.

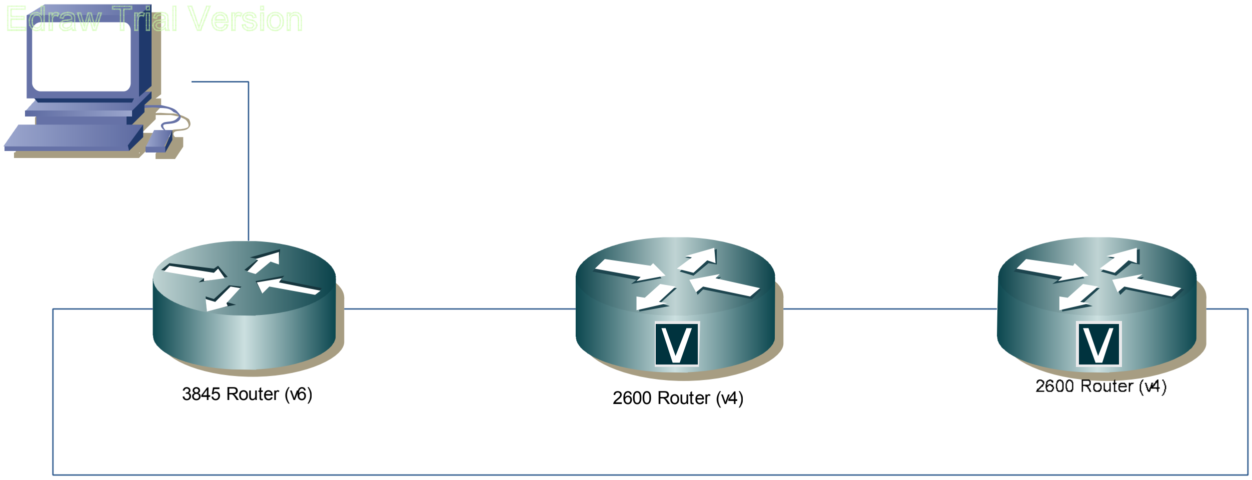


Figure 11

All routers ran RIP as the standard protocol as it was the easiest to implement and represents the "standard" protocol which is run on a wide variety of networks. This enabled easy neighborhood discovery of machines connected to the interfaces. Simple ping and ping6 commands were used to test connectivity between networks. All commands were 100% successful and no loss occurred.

Actual configuration time taken from router setup to test complete was just under three hours total time. The Cisco website was consulted for proper setup instructions and documentation, proving very useful in guiding even a first time user to setup the tunnel.

[5] Encapsulation of IPv6 into IPv4 (and vice versa) packets took place at the 2600 routers

Conclusions

This paper presented two studies which proposed several hypotheses. One hypothesis was not testable, another enabled the collection of data which showed the limits of IPv6, and one presented feasible alternatives to integrating IPv6 into the internet without much pain from the transfer of technology. Many Cisco documents and the earlier RFC's for IPv6 have shown that the "Auto MTU" feature should not pose an issue to networks running the new protocol. End points will take the burden of configuring the initial path MTU discovery while routers will just broadcast their maximum MTU available. While this technology theoretically should work according to plan, there are obviously many unforeseen circumstances which may arise, although none have arisen yet.

Quality of service has generally not been mentioned as an IPv6 feature, although there are numerous references to various proposed projects. A Cisco document shows that QoS is NOT an IPv6 feature (Systems 2003). The protocol will rely on existing protocols, such as TCP and UDP, to keep the same operations as they would in IPv4. While the IPv6 flow label can be used for QoS enabled devices, the label itself is not a QoS feature (Systems 2003).

Address "Auto Configuration" is a very real, working service which has been implemented in IPv6. As witnessed in this research, simply connecting hosts to an already configured router enables that host to learn its address without having to be configured before attachment. The main difference between this service and DHCP is that IPv6 generates link-local and global addresses via a stateless address configuration (Thomson 2007). This configuration happens with minimal configuration of the host router and no additional servers are required to make this happen (Thomson 2007). Since no additional hardware is required, there should be lower costs associated with this type of technology as compared to previous iterations.

The IPv6 to IPv4 tunnel did not take as long as initially thought. In fact, the configuration was so simple to implement that most network architects should have little to no problem making the configuration. All that was required to make it happen was an update of the internal Cisco IOS for one of the 2600 routers (12.2 to 12.3) and a printout of the Cisco configuration document. All the internal tables and routing are taken care of by the IOS and little to no user intervention is required. Enabling RIP, setting the addressing (optional on the V6 router) is all that is necessary to build a separate LAN for a v6 network.

The largest part of this paper, performance metrics on IPv6, was somewhat of a surprise. Initial theories suggested that IPv6 implementation would bring about faster through put, lower latency, and most importantly, a way to see how flexible the protocol is when transferring information from large bandwidth to small bandwidth networks. Unfortunately this was not possible to completely examine due to several outstanding issues.

Before the issues are shown, there were several positive gains from the research. It was determined that there is not discernable difference between changing the MTU value on the host machines to try and simulate small bandwidth networks. In fact, the error rate increased exponentially and the file was still transferred. This was the same for changing the duplex setting to "Half" from the default "Full" or "Auto". It was gathered that given the high propagation speeds possible in modern networking systems, any difference would be so minute that it would not be apparent to the end users. Even changing the values in tandem resulted in a tiny difference in transfer times and was not even worth further analysis.

Speaking to the experts at Force3, a company which specializes in Cisco products and services, it became very apparent that it would not be possible to simulate small bandwidth networks without actually having the hardware capable of those low speeds (Wallace 2008). It was also discovered during this interview that IPv6 is still very much in infancy with a great deal of work ahead in order to fully convert the Internet from IPv4. While many companies had implemented many new devices with v6 capability, there are still many appliances which would not run if switched over to the new protocol.

This brings us back to the 2008 deadline set by the OMB for complete integration of IPv6 into DoD networks. Obviously this has not happened yet as we have less than a month until 2009. Realistically, there are still an inordinate amount of studies which should be conducted before full integration will occur. Too many questions remain to be answered and many older version RFC's for IPv6 have yet to be implemented. However, with the tunneling capability which was tested at the end of this research project, there are many feasible transition options which can occur before full integration sometime in the near future. These transition options will help lessen the strain on resources, such as employee training, which will make the shift cost less to all corporations.

There are still a great deal of interworkings which have to be ironed out before the DoD tries to train military personnel in IPv6. While the concept itself is easy, there are numerous features which are still in varying forms of use which would confuse a great deal of military personnel. On top of this would be the significant cost to train all involved as there would be some intense addressing changes. The biggest concern though comes from the need to swap a greatdeal of hardware in order to be compliant. Many legacy systems still exist today that would not be compatible with the new specifications. A complete redesign of the system would be necessary, all at an enormous expenditure that would be significant to the DoD budget, not to mention at a cost to the American taxpayer.

Finally, there would be the yet to be answered question of implementing IP Security on an IPv6 network. Since there would be no need for NAT or sub netting, theoretically all addresses would be visible to everyone on the global network. While the obvious firewall would be put in place, there are still few applications capable of monitoring those firewalls or even sniffing IPv6 packet traffic.

The eventual transition to IPv6 will happen. There are still large questions that remain to be asked will have to be researched, but unfortunately this all takes a great deal of time, money, and resources. Since a great deal of the civilian sector relies on the DoD to "pave the way" in new technologies, it is with this department that the United States transition to IPv6 lies. Once DoD transfers over to the new protocol, the rest of the country would likely follow.

Results

Technology	**Demonstrated**	**Feasibility**
Dynamic MTU	Possible, but not shown	Possible, use large networks to check
IPv6 to IPv4 Tunnel	Validated	Extremely Feasible

Lessons Learned

This research project produced a great deal of "lessons" which will hopefully be remembered in future initiatives. The largest problem faced in the project dealt with the lack of network resources to test through put. This situation could have been remedied with either the addition of older technologies to incorporate into the lab environment or the use of the IPv6 VPN between the United States Naval Academy and the United States Military Academy. This VPN connection would have more accurately been able to test file transfer in a more realistic environment.

If additional legacy resources would have been utilized, there would have been a greater chance of success of finding messages which dictate the changing MTU sizes for varying link sizes. There would have also been opportunities to setup multiple networks to test the "Plug and Play" feature of IPv6. This would have afforded the project a chance make the lab environment more dynamic. It would have also made it possible to inject more "simulated" traffic to try and mirror what happens across the Internet.

The use of OPNet's Modeler software was useful in helping determine possible baselines and theoretical throughput, but was inaccurate in forecasting throughput. This was possibly due to the various settings that could have been modified within the program. Unfortunately, the company was not able to put out help manuals nor did they offer any useful help on their website. The software also had some minor glitches within the IPv6 module. These issues have been brought to their attention but a fix was not available.

Since IPv6 is a new technology which has not caught on in the mainstream in the United States, there are far fewer resources available in terms of professional training. Online documentation is sparse making it difficult to trust and the relatively few professional sources, Cisco website and the IPv6 official website, make it difficult to tackle the in depth technical issues which arose during the research.

Future Research Opportunities

There are a great deal of research opportunities available for future iterations of this project. One of the prospective goals was the addition of wireless nodes in the mobile network model. Unfortunately, hardware which supported IPv6 was not available and according to the experts at Force3, would not be available since wireless technology is not modified with the addition of IPv6 (Wallace 2008). However, wireless advancements in the near future might make it plausible next year based on increased demand from the civilian and military markets.

In addition to the wireless network, the growing popularity of VoIP dictates the need for a research project aimed solely at VoIP in the IPv6 world. This is also a necessity as military networks are integrating VoIP networks into everyday operations. A possible addition to the VoIP research would be the addition of VoIP security and performance metrics across a mobile network. While IPSec would be possible, the issue still remains for simulating mobile networks when the proper hardware is not available for a lab environment.

Personal Experiences

This research project afforded me an opportunity to learn in many ways. I was able to conduct research and testing into a subject area which was not defined for me, unlike classes which have a strict syllabus. The independent nature of the study allowed me to be as creative as possible with regards to the program of study, network design and implementation, and testing.

One of the largest rewards gained from this experience was working with people ranging from TAD Ensigns to network engineers at ITSD. This wide variety of experience cultivated my own learning and growth. It enabled me to view problems with more insight than previously possible. The ability to work with qualified network engineers and experts in the field was a humbling experience. Valuable insight into the inner workings of the USNA complex was gained and the level of forward thinking was realized.

On a much lower level, simply putting together a research report required an immense amount of time. Fortunately, this report enabled me to learn the process behind writing large technical papers and is a step in the direction of a Masters degree in the near future. The level of detail sometimes required to document the project was overwhelming but very necessary. Overall, the experience of conducting my own research was a delighting experience. The lessons learned from this study will transfer over to my senior project quite easily next semester. There, I will be able to educate my team members with all that I have learned, proving the worth of a semester of research.

Works Cited

Defense, Department of. "Department of Defense Internet Protocol Version 6 Generic Test Plan Version 2." *Defense Information Systems Agency.* 2006.

http://jitc.fhu.disa.mil/adv_ip/register/docs/ipv6_gtp_v2.pdf (accessed November 2008).

Domagalski, Joshua. *Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from Ipv4.* Independent Study Research Project, Annapolis: Computer Science Department, 2007.

Hagen, Silva. *IPv6 Essentials.* Sebastopol: O'Reilly Media, 2006.

Loshin, Pete. *IPv6: Theory, Protocol, and Practice.* San Fransisco: Morgan Kaufmann Publishers, 2003.

Paul, Ryan. *Wubi arrive: A look at Ubuntu 8.04 Alpha 5.* February 24, 2008. http://arstechnica.com/news.ars/post/20080224-wubi-arrives-a-look-at-ubuntu-8-04-alpha-5.html (accessed November 23, 2008).

Systems, Cisco. "Basic IPv6." *Cisco Powered Network Operations Symposium 2003.* San Jose: Cisco Systems, 2003. 47-57.

Thomson, Susan. *IPv6 Stateless Address Autoconfiguration.* San Jose, September 2007.

Vogt, Christian. "A Comprehensive and Efficient Handoff Procedure for IPv6 Mobility Support." *IEEE Computer Society*, 2006: 43-47.

Wallace, interview by 1/C Sean Noronha. *IPv6 Networks* (October 22, 2008).

Zhou, Xiaomind. "Estimation of Preceived Quality of Service for Applications on IPv6 Networks." *Terromolinos*, 2006: 75-77.

Amoss, John. *Handbook of IPv4 to IPv6 Transition: Methodologies for Institutional and Corporate Networks.* Princeton: Auerbach Publications, 2007.